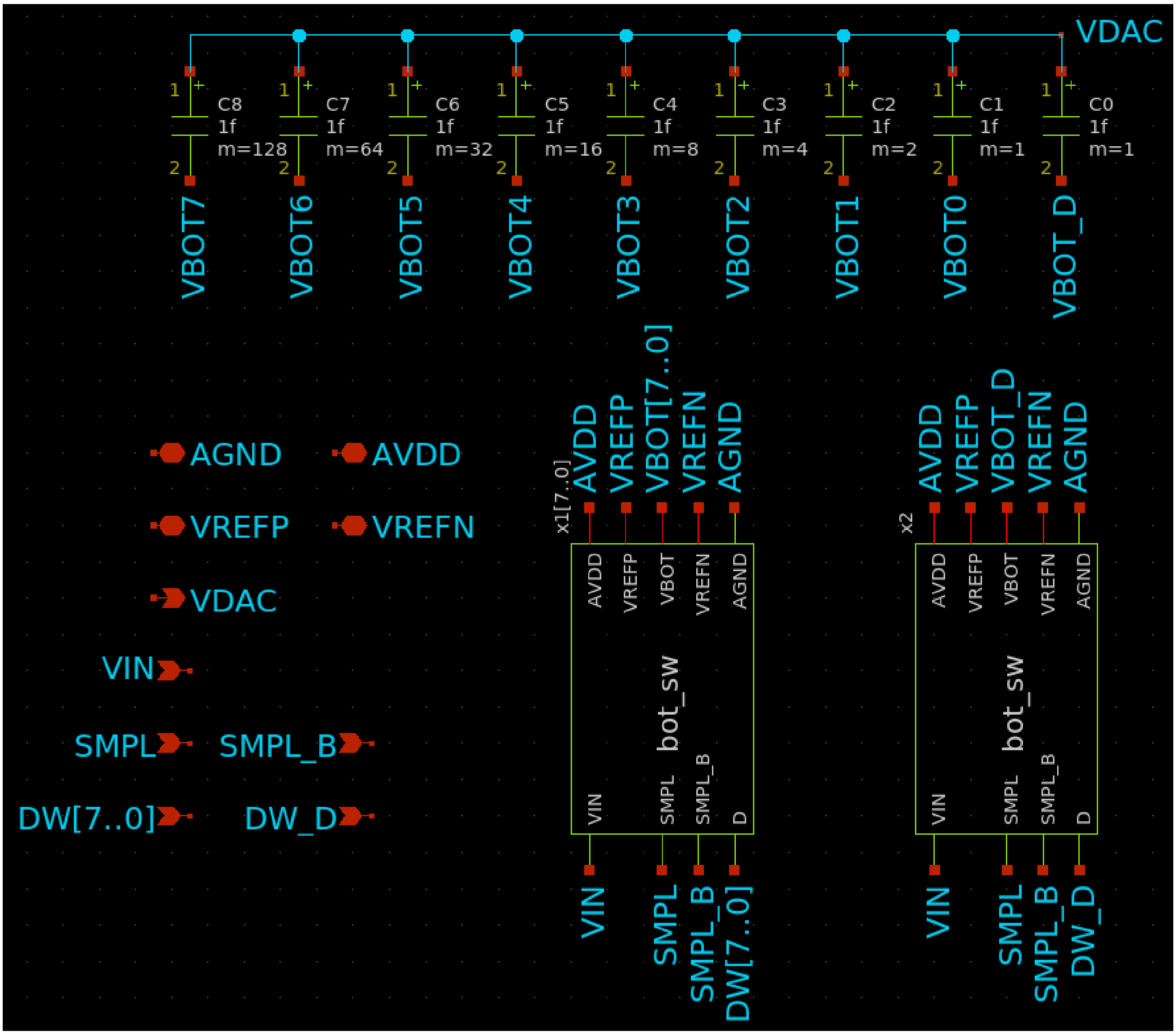
Analog Integrated System Design
Lab 05 – xschem/ngspice
SAR ADC (Mini-Project Part 2)

Intended Learning Objectives

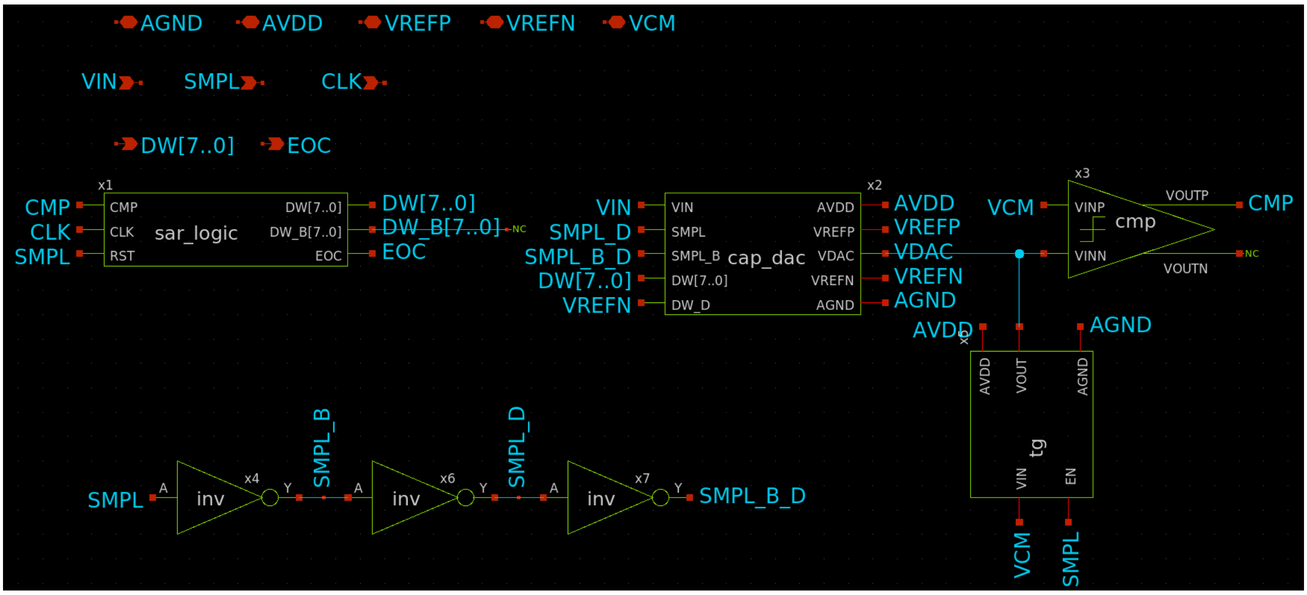
1) To be familiar with behavioral modeling of mixed-signal and digital blocks.
2) To be familiar with the design and simulation of SAR ADC.

Part 1 (Pre-Lab): SAR ADC Design

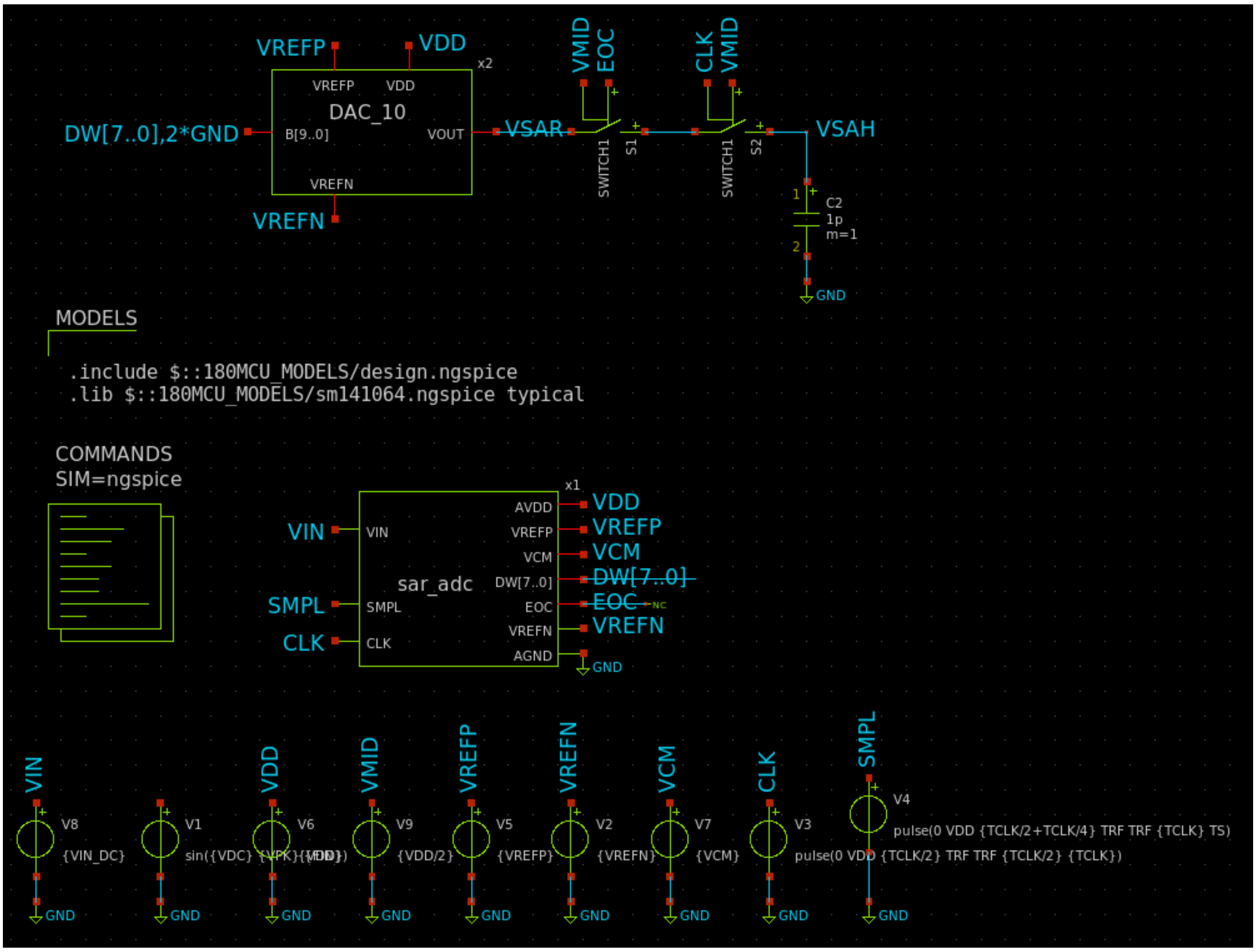
1) Continue working in the same project directory of the previous lab.
2) Create a schematic and a symbol for the capacitive DAC. Note the binary weighted array. The multiplier parameter is used here. But practically, using an array of capacitors is preferred, e.g., C8[127..0].



3) Create a schematic and a symbol for the SAR ADC. Note the delayed version of the sampling clock (SMPL) for correct operation of bottom-plate sampling. DW_B is not used, so it is connected to "devices/noconn" instance to suppress floating net warnings.



4) Create a schematic for the SAR ADC testbench.



```
.include $::180MCU_MODELS/design.ngspice
.lib $::180MCU_MODELS/sm141064.ngspice typical
```

5) Select the input signal source (sine wave or dc) by moving the VIN label from {VIN_DC} to the sine wave source.
6) We use the behavioral DAC_10 to convert the digital output to an analog signal to use it in spectral analysis. Only the 8 MSBs of the DAC are used.
7) Copy the MODELS block to your testbench.
8) Set the ngspice commands as shown below. We use a relatively small number of FFT points to speed up the sine wave test simulation.

```
name=COMMANDS
simulator=ngspice
only_toplevel=false
value="
* ngspice commands

*Required model for the switch (the switch threshold default value is
zero)
* you can define the threshold using vt={VDD/2} (not needed here)
.model SWITCH1 sw

*These are the values of the parameters to be used
.param TS=1u TCLK=100n TRF=1n TDROP={0.5/FIN} TSTOP={(NCYC/FIN)+TDROP}
.param NCYC=5 NFFT=256 FIN={(NCYC/NFFT)/TS}
.param VDD=3 VDC={1} VCM={1} VPK={0.5} VREFP={VCM+VPK} VREFN={VCM-VPK}

.param VCC=VDD
.param VIN_DC={VREFN}
*.param VIN_DC={VREFP}
*.param VIN_DC= {VREFN +(128+32+8+2+0.5)*2*VPK/(2**8)}

*Analysis setup and control statements
*.tran 1n {TSTOP} {TDROP}
.tran 1n {2*TS}

*save all voltages and currents
.save all

*options for an accurate precision output
*.options reltol=1e-6 vntol=1u abstol=1p

* option to make output file ascii
*.options filetype=ascii
"
```

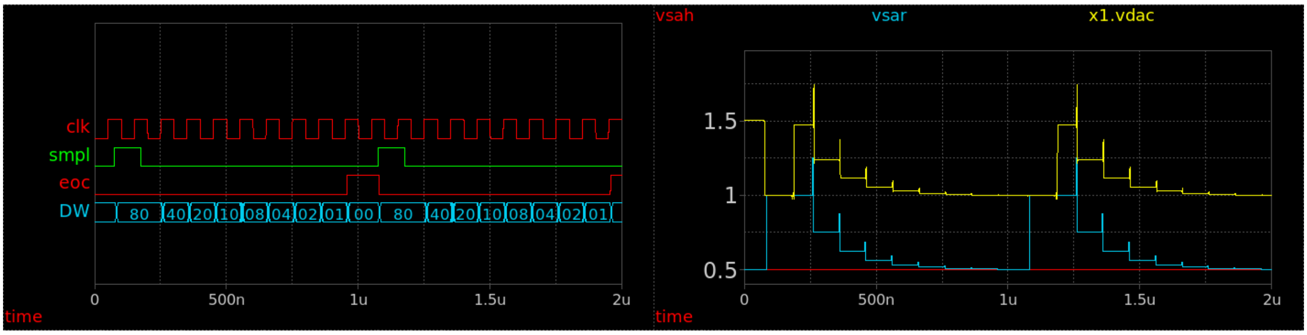
PART 2: DC Functional Test

1) Comment the ascii output option to have a binary output file. Note that waveform graph does not work with ascii output.
2) Configure ngspice in the batch mode to automatically generate the output raw file.
3) Run transient simulation for three cases of VIN by changing the VIN_DC variable in the ngspice commands.
   a. VIN = VREFN → output will be all zeros

b. VIN = VREFN + (128+32+8+2+0.5)*VLSB → output will be alternating zeros and ones

c. VIN = VREFP → output will be all ones

4) Report the waveforms for the three previous cases. An example is shown below. Note the waveforms of VSAR and VDAC.

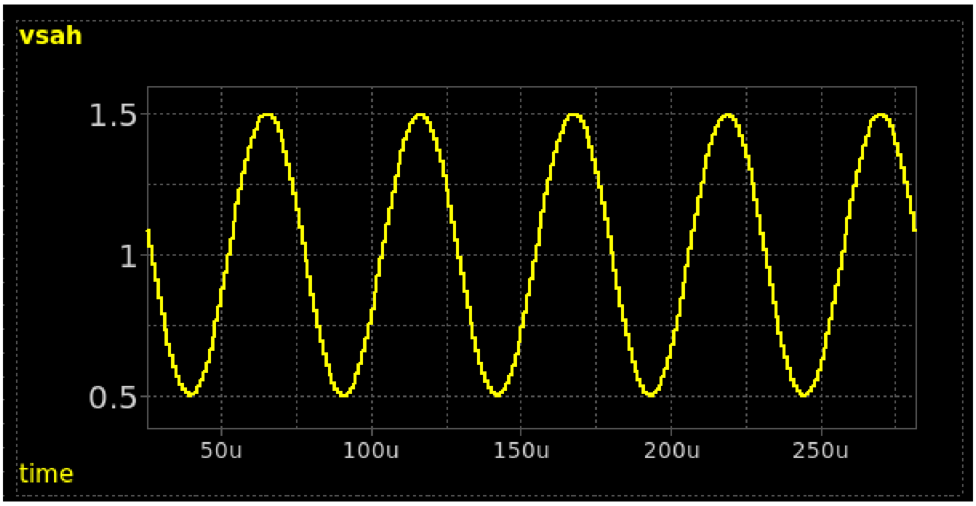


PART 3: Sine Wave Test

1) Run transient analysis for the sine wave test by uncommenting this line in ngspice commands.

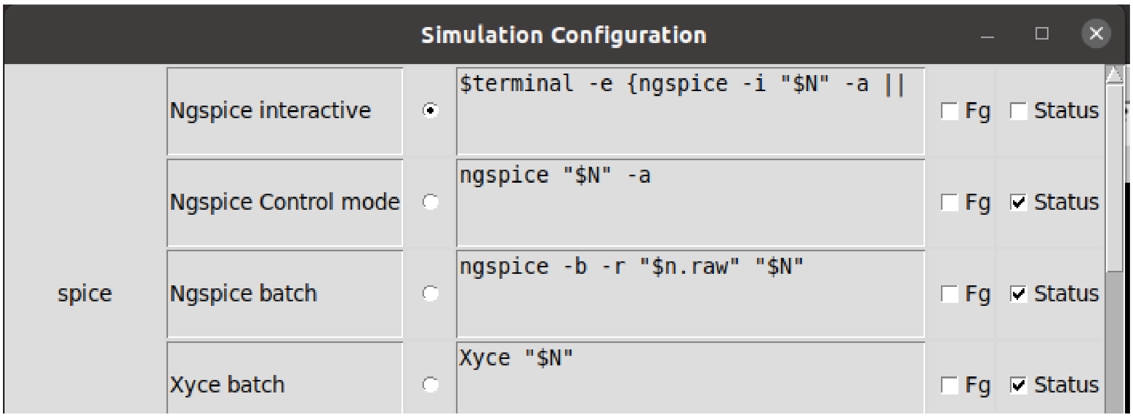
```
.tran 1n {TSTOP} {TDROP}
```

2) Plot transient waveforms of VIN and VSAH **overlaid on the same plot**. VSAH only is shown below as an example.



3) Change the output format to ascii to import it in Octave/Matlab. Uncomment the following ngspice command and configure ngspice in interactive terminal mode.

```
* make output file ascii to import in Octave/Matlab
.options filetype=ascii
```



Simulation Configuration

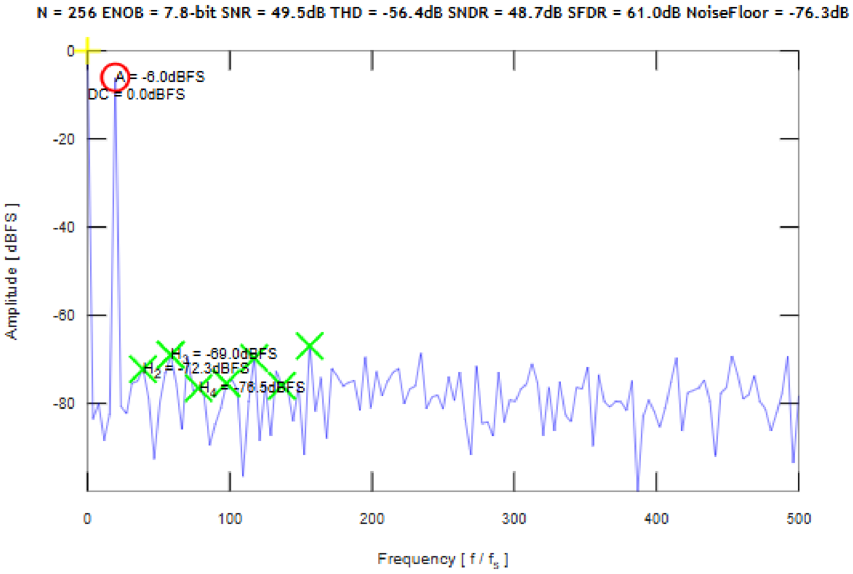
spice	Ngspice interactive	●	`$terminal -e {ngspice -i "$N" -a ||`	□ Fg	□ Status
	Ngspice Control mode	○	`ngspice "$N" -a`	□ Fg	☑ Status
	Ngspice batch	○	`ngspice -b -r "$n.raw" "$N"`	□ Fg	☑ Status
	Xyce batch	○	`Xyce "$N"`	□ Fg	☑ Status

4) In the ngspice terminal write the following commands:

```
let lin-tstep = 1u
linearize v(VSAH)
write vsah_output.raw v(VSAH)
```

5) Use the same procedure and the code in Lab02 to plot the spectrum. Report the following specs. Comment on the results.

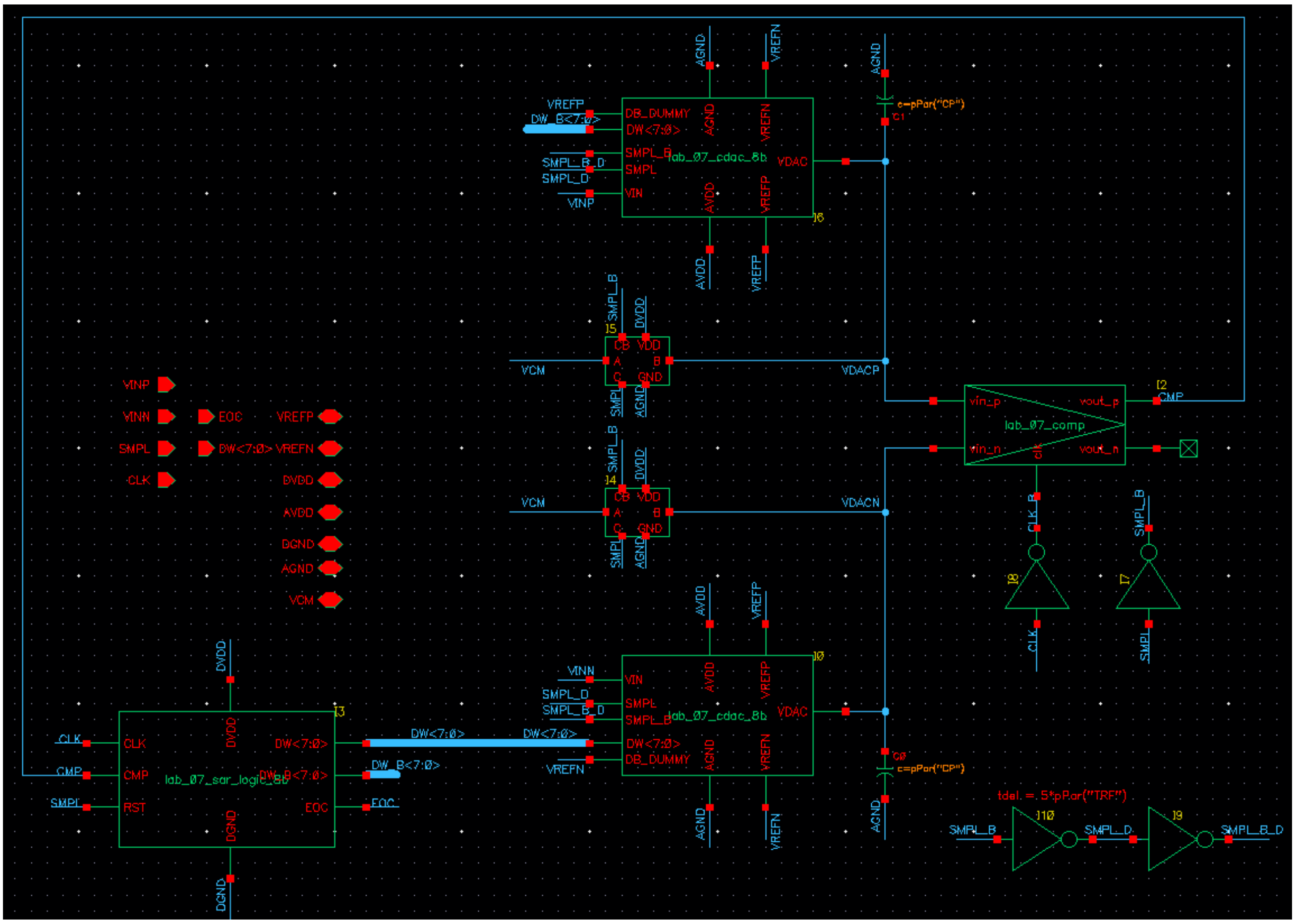
   - ENOB
   - SINAD
   - SNR
   - SFDR
   - THD (in dB)
   - Signal power
   - DC power



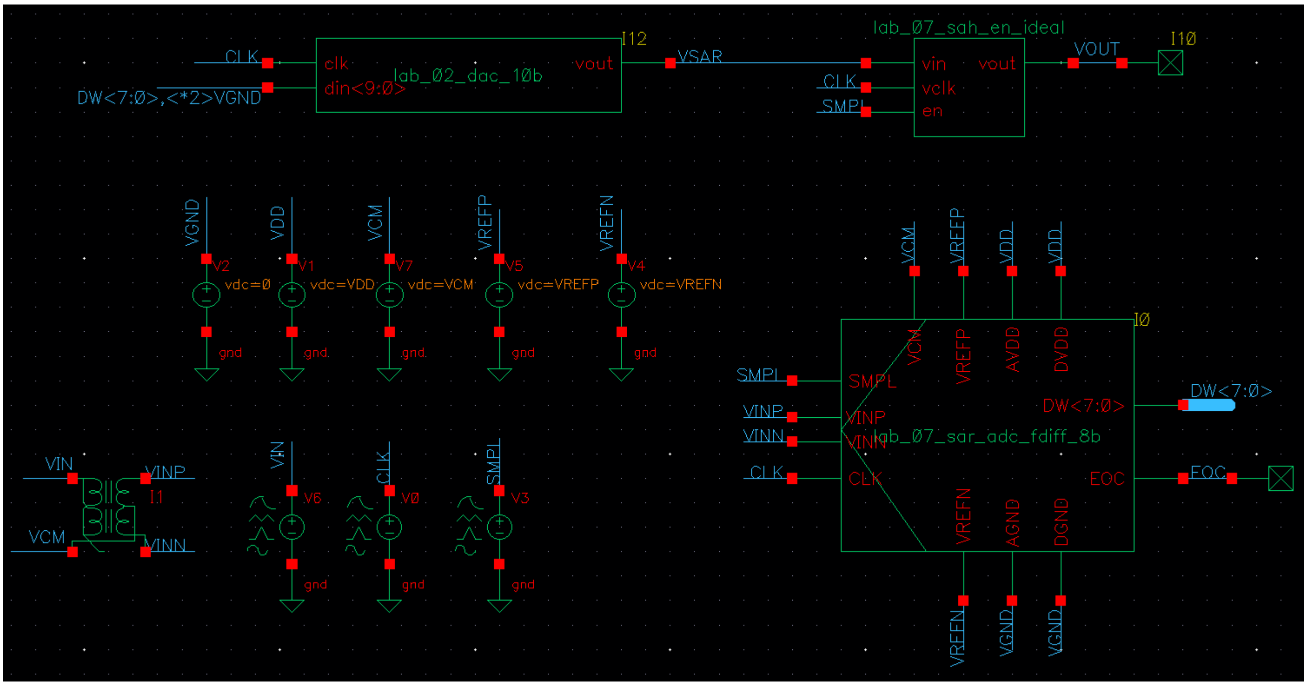
N = 256 ENOB = 7.8-bit SNR = 49.5dB THD = -56.4dB SNDR = 48.7dB SFDR = 61.0dB NoiseFloor = -76.3dB

PART 4 (Bonus): Fully-Differential SAR ADC

1) Create a new schematic for fully-differential SAR ADC. Modify the example schematic below as needed for your design.



2) Create a new schematic for fully-differential SAR ADC testbench. Modify the example schematic below as needed for your design.



3) Switch back to the behavioral comparator (use Hierarchy Editor).

4) Note that the differential operation doubles the amplitude. Remove the DC level for the differential sinusoidal input.

5) Run transient analysis.

6) Plot transient waveforms of VIN and VOUT.
7) Plot the FFT of VOUT to measure the ENOB and other performance parameters.

Acknowledgement

Thanks to all who contributed to these labs. If you find any errors or have suggestions concerning these labs, please contact Hesham.omran@eng.asu.edu.eg.